Systems programming

4 – Interprocess communication

MEEC LEEC MEAer LEAer MEIC-A
João Nuno Silva

TÉCNICO LISBOA

Bibliography

- The Linux Programming interface
  - Chapter 43
- Oracle Solaris 11.4 Programming Interfaces Guide
  - Chapter 6
- Inter-Process Communication (IPC) in Distributed Environments: An Investigation and Performance Analysis of Some Middleware Technologies

System

- Composition of
  - Functions / Modules
  - Classes
  - Processes
- Processes can be running in
  - Different/same space
- Processes can be running at
  - Different/same time

Operating system infrastructure

- Operating systems offer
  - Execution mechanism
  - Protection Mechanisms
  - Communication mechanisms
- Protection
  - Processes are independent entities
    - One process execution does not affect other processes
    - Memory is private

Operating system infrastructure

- Processes in the same system need to exchange information or data:
  - To divide tasks
  - Increase processing power (by distributing tasks into multiple computers/processors)
  - To guarantee synchronization and consistency among them

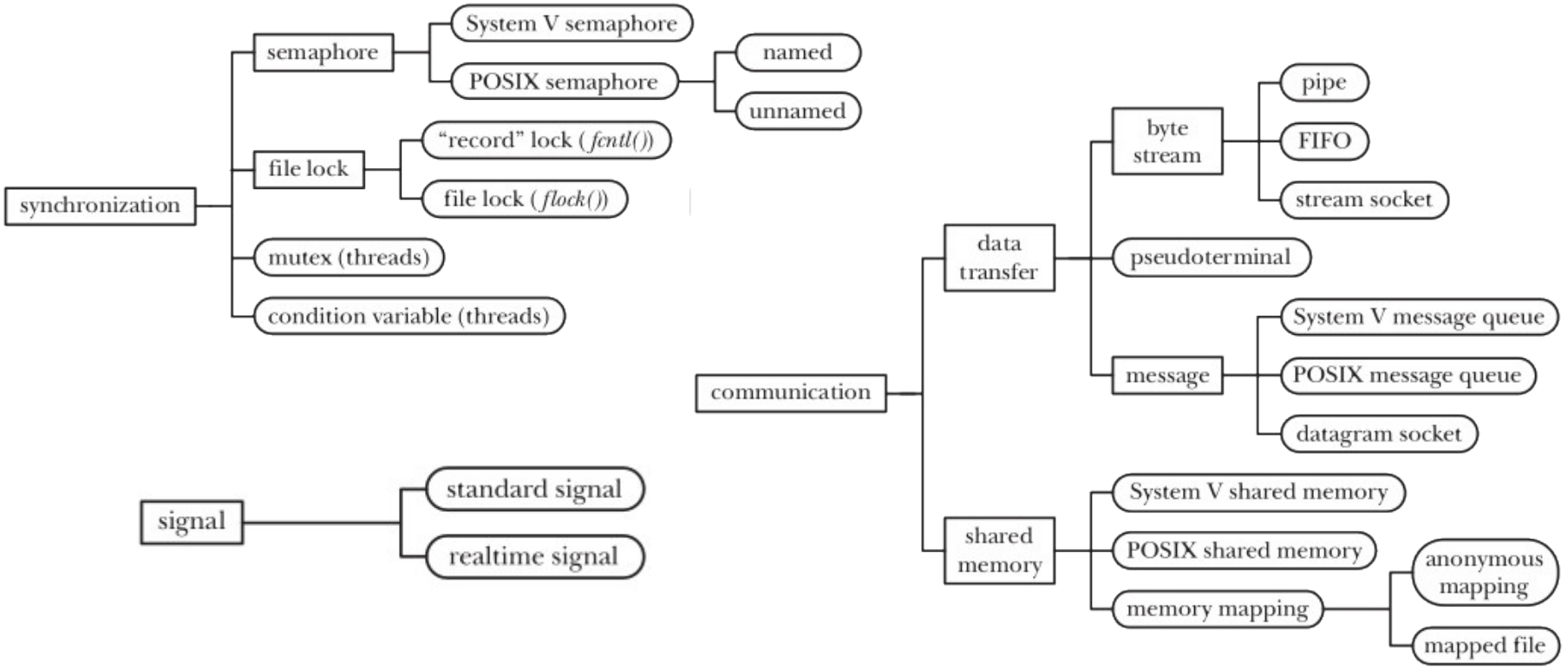
IPC facilities

- Interprocess communication primitives
  - Offered by the operating systems
  - Implemented as a library

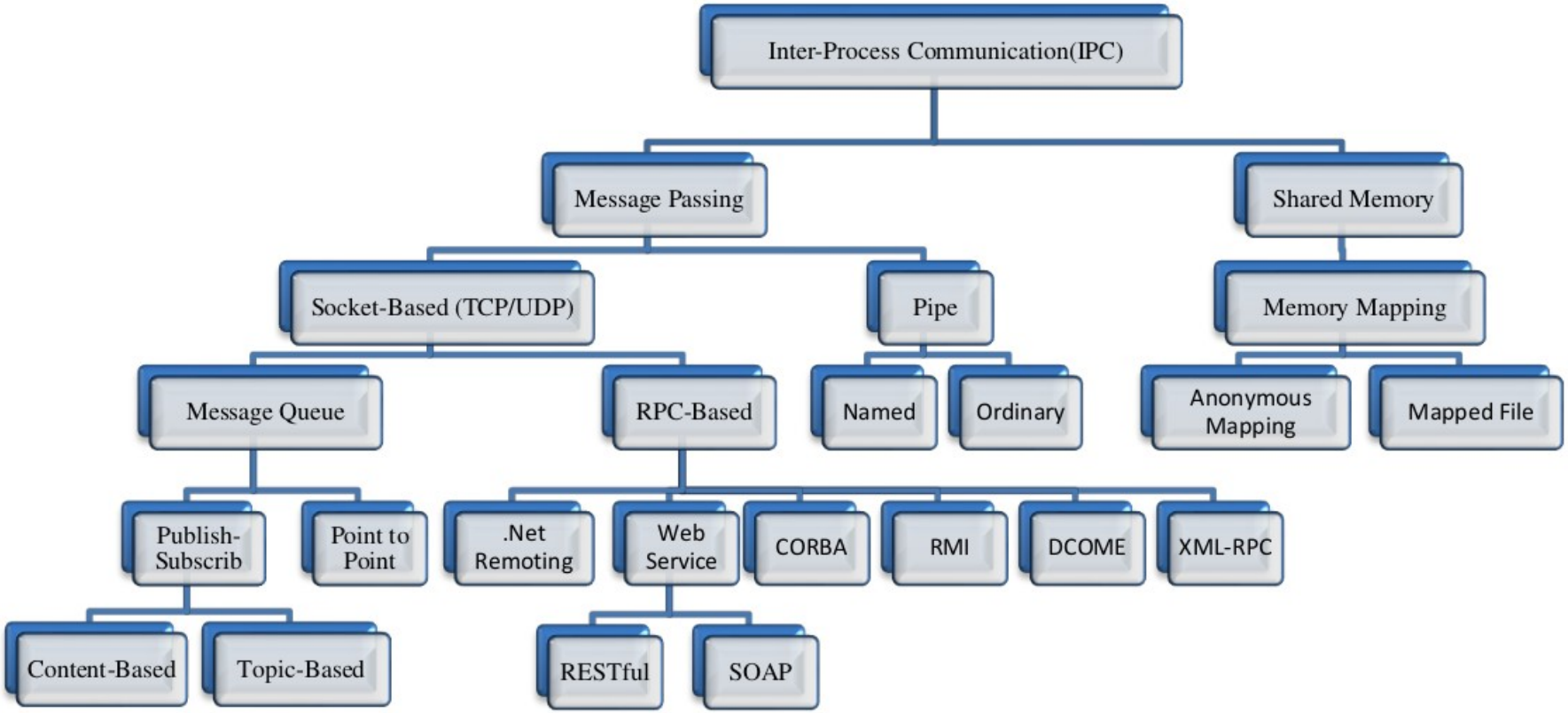
IPC facilities

- Communication
  - concerned with exchanging data between processes.
- Synchronization
  - These facilities are concerned with synchronizing the actions of processes or threads.
- Signals:
  - Event notifications
  - Synchronization technique in certain circumstances.
  - Rarely as a communication technique
    - the signal number itself is a form of information
    - realtime signals can be accompanied by associated data (an integer or a pointer)

IPC facilities



IPC facilities



IPC Characteristics

IPC characteristics

- Different problems require different IPC
  - Explicit vs implicit
  - Implementation
  - Scope
  - Channel identification
  - Handler in program
  - Process relation
  - Time coupling
  - Space coupling

Explicit/Implicit

- Communication can be implicit or explicit
- Explicit communication
  - Required the definition of the message payload
  - Invocation of a send method
  - Invocation of a receive method
- Implicit
  - Shared memory
- Explicit
  - All other mechanims

Implementation

- The transmission of information
  - May require the participation of the Kernel
- Kernel implemented IPC
  - Process call a system call
  - Kernel redirects information to the suitable process
  - Requires data copy
- Middleware implemented IPC

Scope

- What is the location of the entities that communicate
- What is the scope/range of communication
  - In the same process
  - In the same machine
  - In the network
- Scope
  - Local
  - Remote/Distributed

Channel identification

- How are channels identified:
  - Path name / file in the file-systems
  - Network address
  - Key
  - None
- Whats is the relation to scope?

Handler in program

- How are channels identified inside apps?
  - File descriptors
  - Numeric identifiers
  - Pointers
- Pointers
  - Message queues
- Numeric
  - Message queues (other type….)

Process relation

- What processes can communicate
  - Parent Child?
  - Siblings?
  - Any process?
- Related processes
  - Father/sons/brother
- Unrelated processes

- What is the relation with scope?

Time coupling

- Sender and receiver must exist at the same time
  - Or not
  - Related to persistence
  - Related do blocking
- Time uncoupling
  - File System
  - Memory mapped files
  - Message queues

Space coupling

- Related to channel/endpoint identification
- Does participants know the identity of others?
    - Sender knows/specifies who the receiver is
    - Receiver knows how the sender is
  - Or not
- Is it possible to do broadcast
  - Without knowing the identity of the recipients

Characteristics relations

- Scope => Channel identification
- Channel identification => handler in program
- Coupling => Implementation
- Explicit => API
- API => Implementation
- Functionalities => API

IPC Functionalities

- What functionalities offered by the IPC
  - Data organization
  - Duplex communication
  - Accessibility
  - Persistence
  - Blocking
  - Privacy
  - Reliability
  - Ordering

Data organization

- How is data organized in the channel
  - Byte stream
  - Messages

Duplex communication

- Does a single channel offers
  - Half-duplex - One way communication
  - Duplex - Two way communication?
  - None

Accessibility

- Mechanisms to control thge access to the channel
  - Process relation
  - FS permissions / Permission mask
  - None
- Channel in the same machine
  - Related processes
  - Permission mask
- User name/ password?

Persistence

- For how long the IPC object exists/persists
  - A process-persistent IPC object remains in existence
    - as long as it is held open by at least one process
  - A kernel-persistent IPC object remains in existance
    - until either it is explicitly deleted or the system is shut down
  - An file-system persistence IPC object with retains its information
    - even when the system is rebooted

Blocking

- Is access blocking
  - With no other participant
  - With no data
  - Unblocking
- How programs get data?
  - Notification
  - Data Polling

Privacy

- Is communication public?
  - Or private?
- Public communication
  - Any process can participate in the communication
    - Even after establishing of channel
  - Any process can receive the messages
- Private
  - After establishing communication
    - Nobody else can participate

Reliability

- What guarantees are offered
  - Messages are delivered
    - i.e. the receiver can read them
  - Messages are correct

- The kernel implements reliability
- The network may not

Ordering

- Are messages received in the same order as sent?
  - Or not?

- The kernel implements ordering
- The network may not

API

- What is offered by the API
  - Channel creation
  - Channel configuration
  - Data transmission/reception
- It is possible to reuse of other APIS
  - For instance file access

Next on PSIS

- Introduction to processes